# Mehrdimensionale Arrays

- Deklaration Mehrdimensionale Arrays



# Deklaration

- Deklaration eines **zweidimensionalen Arrays:**
  - Datentyp gefolgt von **zwei** aufeinanderfolgenden geöffneten und geschlossenen eckigen Klammern und dem Bezeichner

```
int[][] ganzeZahlenArray;
```

  - Größe des Arrays wird nicht angegeben. Zu diesem Zeitpunkt wird noch kein Speicherplatz für den Array reserviert.

## Deklaration – Eckige Klammer

- Position der eckigen Klammern bei der Deklaration eines **zweidimensionalen Arrays:**
  - Variante 1: - Klammern hinter dem Typ

```
int[][] zweiDimArray01;
```

  - Variante 2: - Klammern hinter dem Bezeichner

```
int zweiDimArray02[][];
```

  - Variante 3: - Klammern hinter dem Typ & dem Bezeichner

```
int[] zweiDimArray03[];
```



## Instantiierung

- Bei der **Instanziierung** wird Speicherplatz reserviert
  - Instanziierung erfolgt mit dem Schlüsselwort ***new***
  - In den eckigen Klammern steht die gewünschte Kapazität des jeweiligen Arrays
  - Kapazitätsangabe für die erste eckige Klammer: **MUSS**

## Instantiierung – erste Art

Kapazitätsangabe für die erste eckige Klammer: **MUSS**

```
ganzeZahlenArray= new int[6][];
```

Innere Arrays sind nicht instantiiert - Kapazitätsangabe im zweiten Klammer nicht vorhanden

```
System.out.println(ganzeZahlenArray);
for (int i = 0; i < ganzeZahlenArray.length; i++) {
    System.out.println(ganzeZahlenArray[i]);
```

```
[[I@64c3c749
null
null
null
null
null
null
```



## Instantiierung – erste Art

Instantiierung der inneren Arrays

```
for (int i = 0; i < ganzeZahlenArray.length; i++) {
    ganzeZahlenArray[i] = new int[i + 1];
    System.out.println(Arrays.toString(ganzeZahlenArray[i]));
}
```

```
[0]
[0, 0]
[0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0]
```

## Instantiierung – erste Art

Kapazitätsangabe in allen Klammern vorhanden

```
ganzeZahlenArray= new int[6][6];
```

Alle Arrays instantiiert

```
System.out.println(ganzeZahlenArray);
for (int i = 0; i < ganzeZahlenArray.length; i++) {
    System.out.println(ganzeZahlenArray[i]);
```

```
[[I@6bbc4459
[0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0]
```



## Instantiierung – zweite Art

Kapazitätsangabe bei direkter Angabe der Elemente führt zu Kompilierfehler

```
ganzeZahlenArray = new int[][]{{1},{1,2},{1,2,3},{1,2,3,4}};
```

Alle Arrays instantiiert

```
for (int i = 0; i < ganzeZahlenArray.length; i++) {
    System.out.println(Arrays.toString(ganzeZahlenArray[i]));
```

```
[1]
[1, 2]
[1, 2, 3]
[1, 2, 3, 4]
```

## Deklaration & Instantiierung – dritte Art

```
int[][] ganzeZahlen= {{1},{1,2},{1,2,3},{1,2,3,4,}};
```

Getrennte Deklaration und Instantiierung führt zu Kompilierfehler

```
int[][] ganzeZahlen;
ganzeZahlen= {{1},{1,2},{1,2,3},{1,2,3,4,}};
```

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
        Array constants can only be used in initializers
```



## Zuweisungen – eindimensionales Array

einDimArray[i] – entspricht einer ganzen Zahl - (i – möglicher Index)

```
int ganzeZahl=10;
int[] einDimArray = new int[]{1,2,3,4,5};
System.out.println(Arrays.toString(einDimArray));

for (int i = 0; i < einDimArray.length; i++) {
    einDimArray[i] += ganzeZahl;
}
System.out.println(Arrays.toString(einDimArray));
```

```
[1, 2, 3, 4, 5]
[11, 12, 13, 14, 15]
```

# Zuweisungen – zweidimensionales Array

zweiDimArray[i] – entspricht eindimensionalem Array

i – möglicher Index)

```
int[] einDimArray = new int[]{1,2,3,4,5};
int[][] zweiDimArray = new int[2][2];

System.out.println(Arrays.deepToString(zweiDimArray) + "\n");

for (int i = 0; i < zweiDimArray.length; i++) {
    zweiDimArray[i] = einDimArray;
}
System.out.println("Nach Zuweisung: zweiDimArray[i] = "
+ "einDimArray \n" + Arrays.deepToString(zweiDimArray));
```



# Zuweisungen – zweidimensionales Array

zweiDimArray[i] – entspricht eindimensionalem Array

i – möglicher Index)

```
[[0, 0], [0, 0]]

Nach Zuweisung: zweiDimArray[i] = einDimArray
[[1, 2, 3, 4, 5], [1, 2, 3, 4, 5]]
```

# Zuweisungen – zweidimensionales Array

zweiDimArray[i][i] – entspricht einer Zahl (i – möglicher Index)

```
int[][] zweiDimArray = new int[5][5];

for (int i = 0; i < zweiDimArray.length; i++) {
    System.out.println(Arrays.toString(zweiDimArray[i]));
}
```

```
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
```



# Zuweisungen – zweidimensionales Array

zweiDimArray[i][i] – entspricht einer Zahl (i – möglicher Index)

```
for (int i = 0; i < zweiDimArray.length; i++) {
    for (int j = 0; j < zweiDimArray[i].length; j++) {
        zweiDimArray[i][j] = (i + 1) * 10 + (j + 1);
    }
}
System.out.println(
        "\nNach Zuweisung: \n"
        + "zweiDimArray[i][j] = " + "(i+1)*10 + (j+1) \n");

for (int i = 0; i < zweiDimArray.length; i++) {
    System.out.println(Arrays.toString(zweiDimArray[i]));
```

## Zuweisungen – zweidimensionales Array

zweiDimArray[i][i] – entspricht einer Zahl (i – möglicher Index)

```
Nach Zuweisung:
zweiDimArray[i][j] = (i+1)*10 + (j+1)

[11, 12, 13, 14, 15]
[21, 22, 23, 24, 25]
[31, 32, 33, 34, 35]
[41, 42, 43, 44, 45]
[51, 52, 53, 54, 55]
```



## Zuweisungen – drei dimensionales Array

dreiDimArray[i] – entspricht zweidimensionalem Array
i – möglicher Index)

```
int[][] zweiDimArray = new int[][]{{6,7},{8,9}};
int[][][] dreiDimArray = new int[2][2][2];

System.out.println(Arrays.deepToString(dreiDimArray) + "\n");

for (int i = 0; i < dreiDimArray.length; i++) {
    System.out.println(Arrays.deepToString(dreiDimArray[i]));
```

```
[[[0, 0], [0, 0]], [[0, 0], [0, 0]]]

[[0, 0], [0, 0]]
[[0, 0], [0, 0]]
```

# Zuweisungen – drei dimensionales Array

dreiDimArray[i] – entspricht zweidimensionalem Array

i – möglicher Index)

```
for (int i = 0; i < dreiDimArray.length; i++) {
    dreiDimArray[i] = zweiDimArray;
}
System.out.println("\nNach Zuweisung: \n"
        + "dreiDimArray[i] = zweiDimArray \n");

for (int i = 0; i < dreiDimArray.length; i++) {
    System.out.println(Arrays.deepToString(dreiDimArray[i]));
```

```
Nach Zuweisung:
dreiDimArray[i] = zweiDimArray

[[6, 7], [8, 9]]
[[6, 7], [8, 9]]
```



# Zuweisungen – drei dimensionales Array

dreiDimArray[i][i] – entspricht eindimensionalem Array

i – möglicher Index)

```
int[][][] dreiDimArray = new int[2][2][2];

System.out.println(Arrays.deepToString(dreiDimArray) + "\n");

for (int i = 0; i < dreiDimArray.length; i++) {
    System.out.println(i + ": "
    + Arrays.deepToString(dreiDimArray[i]) + "\n");
    for (int j = 0; j < dreiDimArray[i].length; j++) {
        System.out.println(i + "," + j + ": "
    + Arrays.toString(dreiDimArray[i][j]));
```

# Zuweisungen – drei dimensionales Array

dreiDimArray[i][i] – entspricht eindimensionalem Array

i – möglicher Index)

```
[[[0, 0], [0, 0]], [[0, 0], [0, 0]]]


0: [[0, 0], [0, 0]]

0,0: [0, 0]
0,1: [0, 0]

1: [[0, 0], [0, 0]]

1,0: [0, 0]
1,1: [0, 0]
```



# Zuweisungen – drei dimensionales Array

dreiDimArray[i][i] – entspricht eindimensionalem Array

```
for (int i = 0; i < dreiDimArray.length; i++) {
    for (int j = 0; j < dreiDimArray[i].length; j++) {
        dreiDimArray[i][j] = einDimArray;
    }
}

System.out.println("\nNach Zuweisung: \n"
+ "dreiDimArray[i][j] = einDimArray \n");

System.out.println(Arrays.deepToString(dreiDimArray) + "\n");

for (int i = 0; i < dreiDimArray.length; i++) {
    System.out.println("\n" +i + ": "
        +   Arrays.deepToString(dreiDimArray[i]) + "\n");
    for (int j = 0; j < dreiDimArray[i].length; j++) {
        System.out.println(i + "," + j + ": "
    + Arrays.toString(dreiDimArray[i][j]));
```

# Zuweisungen – drei dimensionales Array

dreiDimArray[i][i] – entspricht eindimensionalem Array

```
Nach Zuweisung:
dreiDimArray[i] = zweiDimArray

[[[1, 2, 3, 4, 5], [1, 2, 3, 4, 5]], [[1, 2, 3, 4, 5], [1, 2, 3, 4, 5]]]


0: [[1, 2, 3, 4, 5], [1, 2, 3, 4, 5]]

0,0: [1, 2, 3, 4, 5]
0,1: [1, 2, 3, 4, 5]

1: [[1, 2, 3, 4, 5], [1, 2, 3, 4, 5]]
```



# Zuweisungen – drei dimensionales Array

dreiDimArray[i][i][i] – entspricht einer Zahl

```
int[][][] dreiDimArray = new int[2][2][2];

System.out.println(Arrays.deepToString(dreiDimArray) + "\n");

for (int i = 0; i < dreiDimArray.length; i++) {
    System.out.println("\n" + i + ": "
    + Arrays.deepToString(dreiDimArray[i]) + "\n");
    for (int j = 0; j < dreiDimArray[i].length; j++) {
        System.out.println(i + "," + j + ": "
        + Arrays.toString(dreiDimArray[i][j]));
        for (int k = 0; k < dreiDimArray[i][j].length; k++) {
            System.out.println(i + "," + j + "," + k + ": "
            + dreiDimArray[i][j][k]);
```

# Zuweisungen – drei dimensionales Array

dreiDimArray[i][i][i] – entspricht einer Zahl

```
[[[0, 0], [0, 0]], [[0, 0], [0, 0]]]

0: [[0, 0], [0, 0]]

0,0: [0, 0]
0,0,0: 0
0,0,1: 0
0,1: [0, 0]
0,1,0: 0
0,1,1: 0
1: [[0, 0], [0, 0]]

1,0: [0, 0]
1,0,0: 0
1,0,1: 0
1,1: [0, 0]
1,1,0: 0
1,1,1: 0
```